

SC2006: Software Engineering

Lab 3 Deliverables

DateGoWhere

Team SPAZM

Lu Zihao (U2122553H)

Markus Lim En Cheng (U2222471K)

Paviru Vidwan Perera (U2222381L)

Tam Yik Lock (U2222001C)

Tan Shao Chong (U2222070J)

Table of Contents

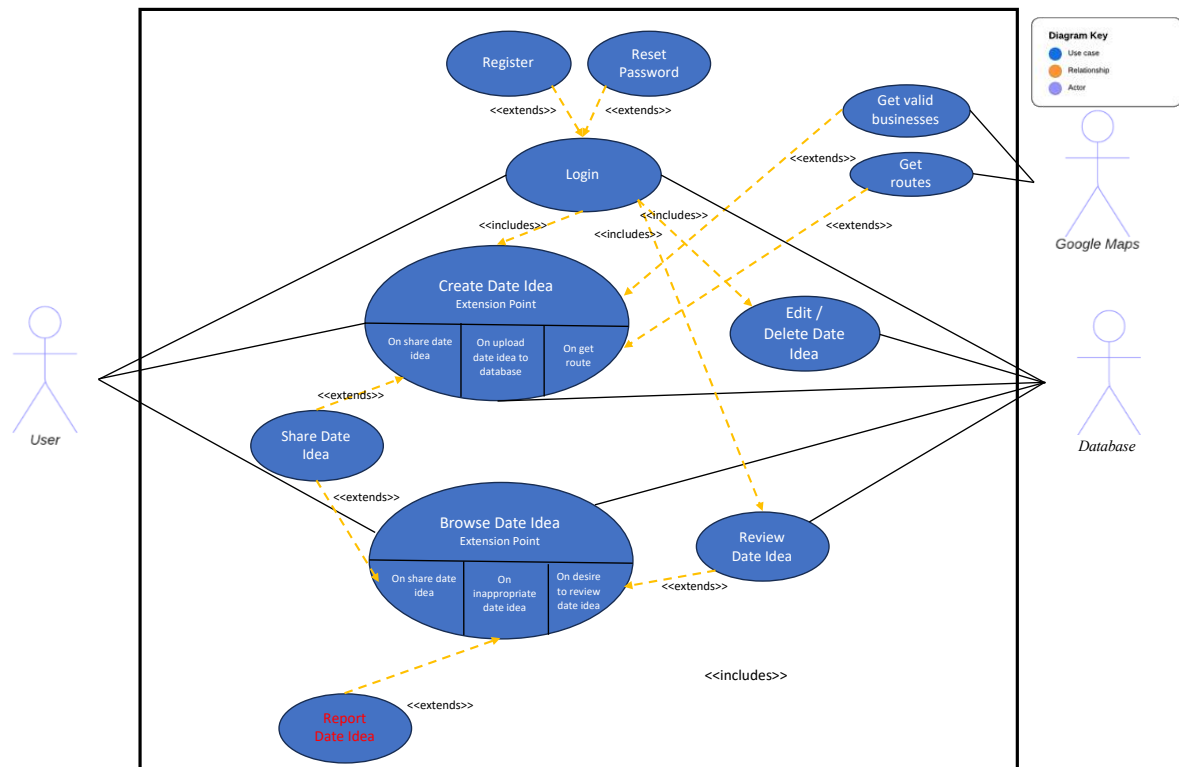
Introduction	3
Use Case Model	4
Use Case Diagram	4
Use Case Descriptions	5
Class Diagram	21
Sequence Diagrams	22
Login	22
Register	22
Reset Password	23
Browse Ideas	23
Create Idea	24
Review Idea	24
Report Idea	25
Dialog Map	26
System Architecture	27
App Skeleton	28
Landing	28
Authentication	28
Create Date Idea	29
View Date Idea	30
Review Date Ideas	31
Browse Date Idea	32

Introduction

DateGoWhere is a web application that facilitates the planning of dates with significant others, families, and friends. Users can share their date itineraries on the app's catalogue for others to view and use. Users can also rate other date itineraries in the catalogue, as well as leave their reviews.

Use Case Model

Use Case Diagram



Use Case Descriptions

Use Case ID:	1.1		
Use Case Name:	Login User		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database)
Description:	All Users must log in to their account before they can create ideas or review ideas. This is to provide legitimacy and accountability of ideas uploaded and the quality of reviews.
Preconditions:	<ol style="list-style-type: none"> 1. User has an existing account AND 2. have not logged in.
Postconditions:	User successfully logs in.
Priority:	High. User will need to login to his/her account to be able to review, report or upload date ideas.
Frequency of Use:	1 time per session
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs their email and password in the respective input boxes or logs in via Google Account. 2. User clicks "Login". 3. System submits a POST request with the user's credentials for verification with the database. 4. System checks for existence of email. 5. System validates password associated with the user's email. 6. System displays "Successful sign in". 7. System redirects the user to the home page
Alternative Flows:	<p>AF1-S4: Email does not exist:</p> <ol style="list-style-type: none"> 1. System displays "Invalid email, please register" 2. System returns to step 1. User can choose to try login again or click Register instead. <p>AF2-S5: Invalid password:</p> <ol style="list-style-type: none"> 1. System displays "Invalid password, if you forgot your password, please reset" 2. System returns to step 1. User can choose to login with a different password or click Reset Password instead.
Exceptions:	EX-S5 – If User inputs password incorrectly more than 5 times, the app displays message "Account suspended". A reset password email is sent to the registered email, for the User to login again.
Includes:	-
Special Requirements:	User has a registered account in the app database.

Assumptions:	-
Notes and Issues:	-

Use Case ID:	1.2		
Use Case Name:	Register User		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	13/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database)
Description:	If a user does not have an account registered with the app, he is required to register using an email to enable him to login for use of review, report and uploading of date ideas.
Preconditions:	1. User does not have an existing account
Postconditions:	User successfully registers an account.
Priority:	Medium. User will need to register an account to gain access to features within the app.
Frequency of Use:	1 time per user lifetime.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks "Don't have an account? Register" on Entry Page. 2. User inputs "Email" and "Password" respective input boxes or registers via Google Account. 3. User clicks "Register" button. 4. System submits a POST request with the user's email the database for verification with the database. 5. If email does not exist on the database, system performs a check on the password input to determine whether it satisfies the conditions: <i>at least 8 characters long AND contains at least one uppercase (A-Z) AND one lowercase (a-z) character, AND any of the following special characters: (!, @, #, \$, %, ^, &, *, (,), _ , -, =, +).</i> 6. If password satisfies the conditions, system submits a POST request with the user's email and password to add into the database. 7. System will prompt user "Signed-up successfully" upon successful registration. 8. The system redirects the user to the login page
Alternative Flows:	<p>AF3-S5: Email already exists</p> <ol style="list-style-type: none"> 1. System prompts user: "Email is already used, please login or choose another email". 2. System returns to step 1. User can either sign up with a different email or go back to login with current email. <p>AF1-S6: Password does not meet requirement</p> <ol style="list-style-type: none"> 1. System prompts users: "Please ensure that your password consist of at least 8 characters long AND

	contains at least one uppercase (A-Z) AND one lowercase (a-z) character, AND any of the following special characters: (!,@,#,\$,%,^,&*,(,),_,-,=,+)." 2. System returns to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	User email is not registered in the database.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	1.3		
Use Case Name:	Reset password		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	13/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database)
Description:	If a user is unable to login to his account, he can reset his password using his email
Preconditions:	<ol style="list-style-type: none"> 1. User has an existing account AND <ol style="list-style-type: none"> 2. User is not signed in
Postconditions:	User successfully resets password to account.
Priority:	Low. Only users who has forgotten their password will use this feature.
Frequency of Use:	2-3 times per user lifetime.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks "Reset Password" on Entry Page. 2. User inputs "Email" in input box. 3. If email exists in the database, system sends an email to user's email. 4. User clicks on the hyperlink in the email. 5. User is directed to "Password Reset" page 6. User inputs "New Password" in input box. 7. User clicks on "Confirm Password Reset" button. 8. System performs a check on the password input by the user and satisfy the conditions: <i>at least 8 characters long AND contains at least one uppercase (A-Z) AND one lowercase (a-z) character, AND any of the following special characters: (!, @, #, \$, %, ^, &, *, (,), _, -, =, +).</i> 9. If password is valid, system submits a POST request with the user's email and new password for verification with the database. 10. System will successfully modify password in the database. 11. The system redirects the user to the login page
Alternative Flows:	<p>AF-S3: Email does not exist.</p> <ol style="list-style-type: none"> 1. System prompts user: "Email not found please sign up" 2. Redirects user to sign-up page. <p>AF-S9: Password does not meet requirement.</p> <ol style="list-style-type: none"> 1. System prompts user: "Please ensure that your password satisfy the necessary conditions". 2. Returns to step 1.

Exceptions:	-
Includes:	-
Special Requirements:	User email is registered in the database.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2.1		
Use Case Name:	Create Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database), Google Maps API
Description:	User can upload their date ideas onto the app for others to view. They can include information such as location, time, price, and activities to be done.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in and authenticated <p style="text-align: center;">AND</p> <ol style="list-style-type: none"> 2. User is at homepage.
Postconditions:	Date idea is successfully uploaded to the database.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Create New Idea" option. 2. User fills in date name, category tags, and whether date is set to public or private. 3. User adds activities to the date idea via individual activity forms. 4. The app processes the various activity locations and calculates the route using Google Places and Routes API. 5. User submits the date idea. 6. The app processes the information provided. 7. If uploading is successful, the date idea is added to the app's database. 8. The app notifies the user of successful upload.
Alternative Flows:	AF-S6. If uploading fails (e.g., missing information), the app displays an error message, and the user is prompted to correct the details. App returns to Step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to uploading date idea.
Special Requirements:	-

Assumptions:	Each location provided for activities are valid locations in Google Places API
Notes and Issues:	-

Use Case ID:	2.2		
Use Case Name:	Browse Date Ideas		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database)
Description:	User can browse various public date ideas. Users can filter date ideas by category tags, keyword and price. Users can select date idea to view the location route, price and activities done during the date.
Preconditions:	User is at homepage.
Postconditions:	User views a list of date ideas based on applied filters and views further information on interested date idea.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Browse Ideas" option. 2. User applies filters (e.g., price, keyword, category tag) to narrow down the list of date ideas. 3. The app retrieves and displays date ideas based on the applied filters. 4. User can click on a date idea to view more information. 5. User views a map feature with location of date idea and a list of information such as price and activities to be done.
Alternative Flows:	AF-S3. If no date ideas match the applied filters, the app informs the user accordingly. App returns to Step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Get public date ideas from database to be displayed.
Special Requirements:	-
Assumptions:	There are date ideas uploaded to the database.
Notes and Issues:	-

Use Case ID:	2.3		
Use Case Name:	Share Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor)
Description:	User can share the date idea with partner or friends. The sharing will be done through social media platforms such as Telegram, WhatsApp, or Email.
Preconditions:	-
Postconditions:	User shares the date idea to their intended recipient.
Priority:	Medium
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea from their own created ideas or public date ideas they want to share. 2. User chooses the "Share" option. 3. The app provides options for sharing via various social media platforms. 4. User selects a platform and shares the date idea with their partner or friends. 5. The app processes the sharing. 6. The app notifies the user that the idea has been shared successfully.
Alternative Flows:	AF-S5. If there are issues with sharing on social media (e.g., authentication failure), the app informs the user. App returns to Step 4.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2.4		
Use Case Name:	Review Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	24/10/23

Actor:	User (Initiating Actor), Database (User Account Database)
Description:	User can review a date idea after going through it. User can provide a star rating out of 5 and also type out a review paragraph of maximum.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in and authenticated <p><i>AND</i></p> <ol style="list-style-type: none"> 2. User is at a date idea information page.
Postconditions:	User submits a review for the date idea.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they have completed. 2. User chooses the "Add review" option. 3. User provides a rating out of 5 stars. 4. User can include a written review of the date idea. 5. User submits his review to be posted publicly. 6. The app processes the review. 7. The app notifies the user that the review has been successfully submitted.
Alternative Flows:	AF-S6. If review cannot be processed (e.g. Too many characters for review, the app will display an error message. App returns to Step 5.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to reviewing date idea.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 002		
Use Case Name:	Delete/Edit Date Idea		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can edit or delete their uploaded date ideas. Date ideas specific to events may be outdated or unavailable, hence ideas must be updated to remain relevant.
Preconditions:	User is logged in and authenticated.
Postconditions:	Date idea is successfully deleted or edited.
Priority:	Low
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they have previously uploaded. 2. User chooses to either edit or delete the date idea. 3. If editing, the user makes changes to the date idea details and submits the updated information. 4. If deleting, the user confirms to delete the idea. 5. The app processes the edit or delete request. 6. The app notifies the user of the success or failure of the request.
Alternative Flows:	AF-S5. If processing fails (e.g., missing information), the app displays an error message, and the user is prompted to correct the details. App returns to Step 3 or 4.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication required prior to modifying date idea.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCX - 004		
Use Case Name:	Plan Date Idea (*REMOVE)		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	User
Description:	User can select interested date idea and input their home address, partner's address, and other locations he/she wants to visit. The app will then suggest the fastest route based on Google Maps route API.
Preconditions:	User has app downloaded.
Postconditions:	User receives a planned route with estimated time required.
Priority:	Medium
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a date idea they are interested in. 2. User chooses the "Plan Date Idea" option. 3. User inputs the locations they plan to visit during the date, in sequence. 4. The app uses the Google Maps API to calculate and display an end-to-end route from the user's location to the date idea's location and other included locations. 5. The app estimates the time required for the planned date. 6. User can choose to discard, save or share the planned route.
Alternative Flows:	AF-S4. If the Google Maps API is unable to provide location for the User input location, an error message will be displayed. App returns to Step 3.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Google Maps Routes API to plan routes. 2. Google Maps Places API to get valid businesses 3. Share Date Idea is extended to provide option for User to share route with their partner.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	Time estimation will be based on expected duration at the intended time of date. Exact time will be unable as User will likely be planning date beforehand.

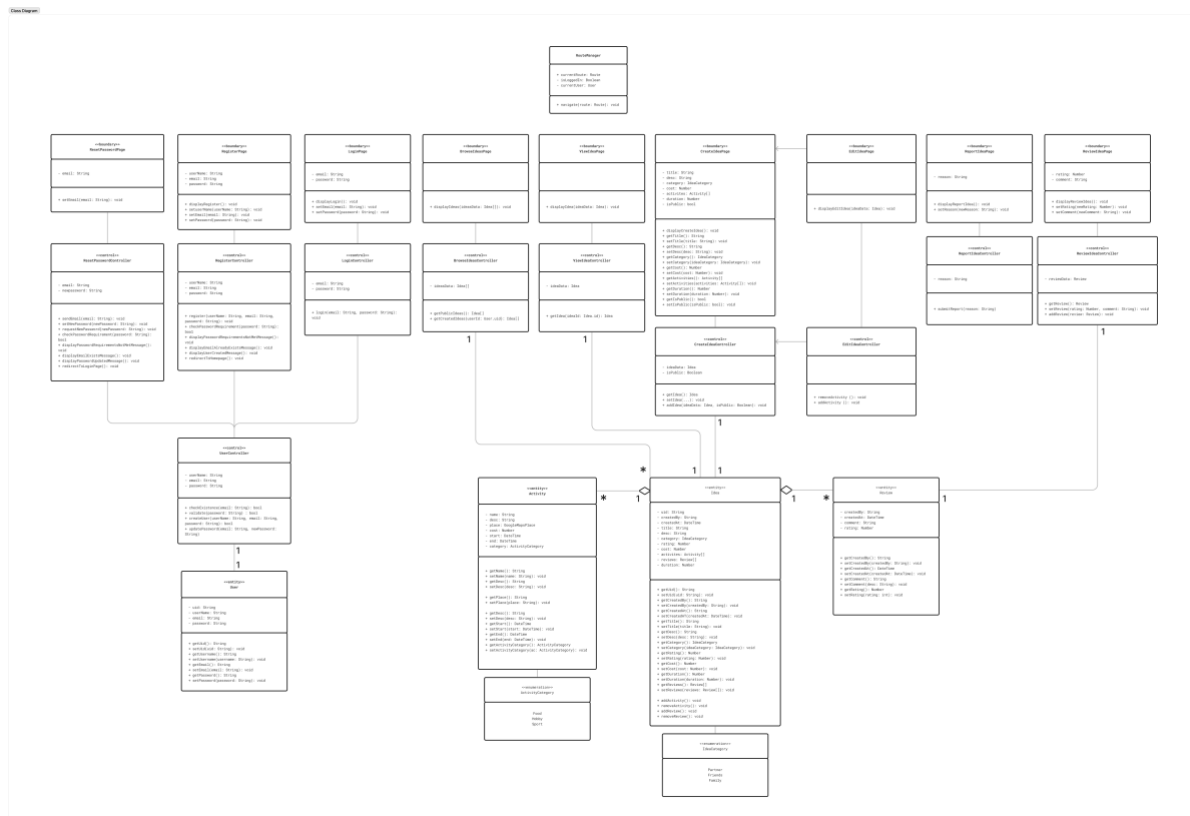
Use Case ID:	UCG - 001		
Use Case Name:	Get Valid Businesses		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	Google Maps
Description:	Retrieve location information for an input address on the integrated Map feature. This will help users to plan their date more easily.
Preconditions:	User has input an address/postal code/place name into the App.
Postconditions:	User is successfully displayed input location on the Map.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs the location either through text input or allowing access to location data. 2. App provides input to Google Maps Places API. 3. If Google Maps Location API can find location, it provides details of location to App. 4. App displays location as a marker on the Integrated Map.
Alternative Flows:	AF-S3 - If Google Maps Places API cannot find location (e.g., invalid postal code), the app displays an error message and prompts the user to enter another location. App returns to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	Google Maps Places API
Assumptions:	-
Notes and Issues:	-

Use Case ID:	UCG - 002		
Use Case Name:	Get Routes		
Created By:	Paviru	Last Updated By:	Paviru
Date Created:	03/09/23	Date Last Updated:	03/09/23

Actor:	Google Maps
Description:	Retrieve optimal route for an input address and location listed on date idea on the integrated Map feature. This will help users to plan their date more easily.
Preconditions:	User has input at least 2 address/postal code/place name into the App.
Postconditions:	User is successfully displayed route going through all input locations on the Map.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs multiple locations either through text input or allowing access to location data. User must put the locations in sequence of intended visit. 2. App provides input to Google Maps Routes API. 3. If Google Maps Routes API can find all locations and valid routes to all locations, it provides details of route to App. 4. App displays route on the Integrated Map.
Alternative Flows:	AF-S3 - If Google Maps Routes API cannot find location (e.g., invalid postal code), the app displays an error message and prompts the user to enter another location. App returns to Step 2. - If Google Maps Routes API cannot find a valid route (e.g., inaccessible location), the app displays an error message and prompts the user to enter another location. App returns to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	Google Maps Location API
Assumptions:	-
Notes and Issues:	-

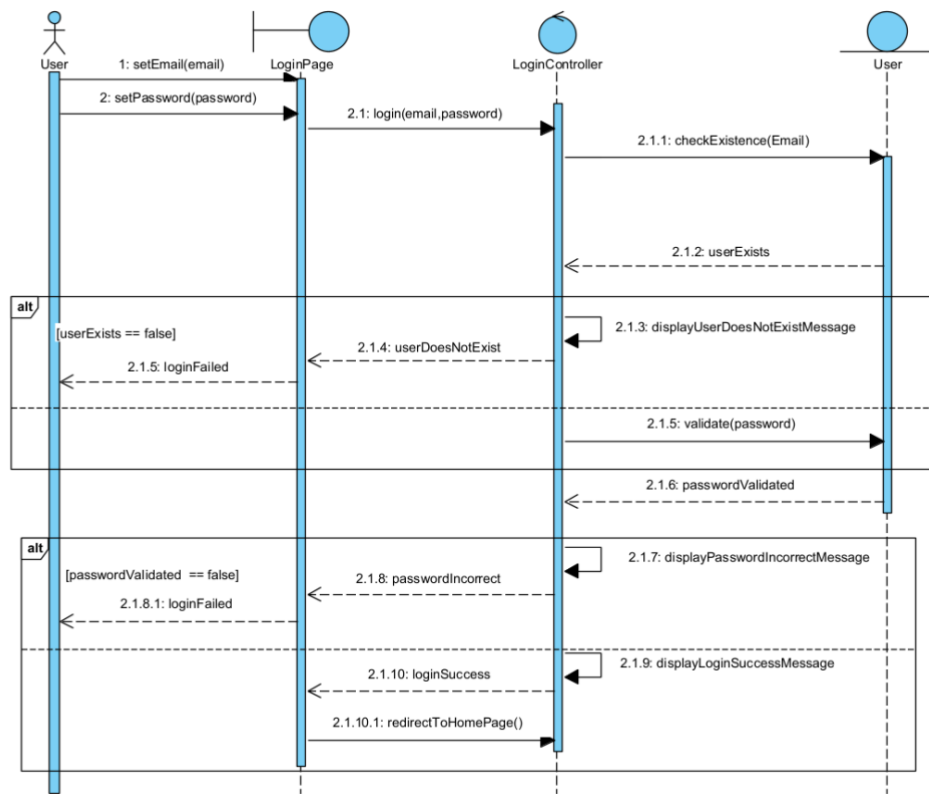
Class Diagram



Sequence Diagrams

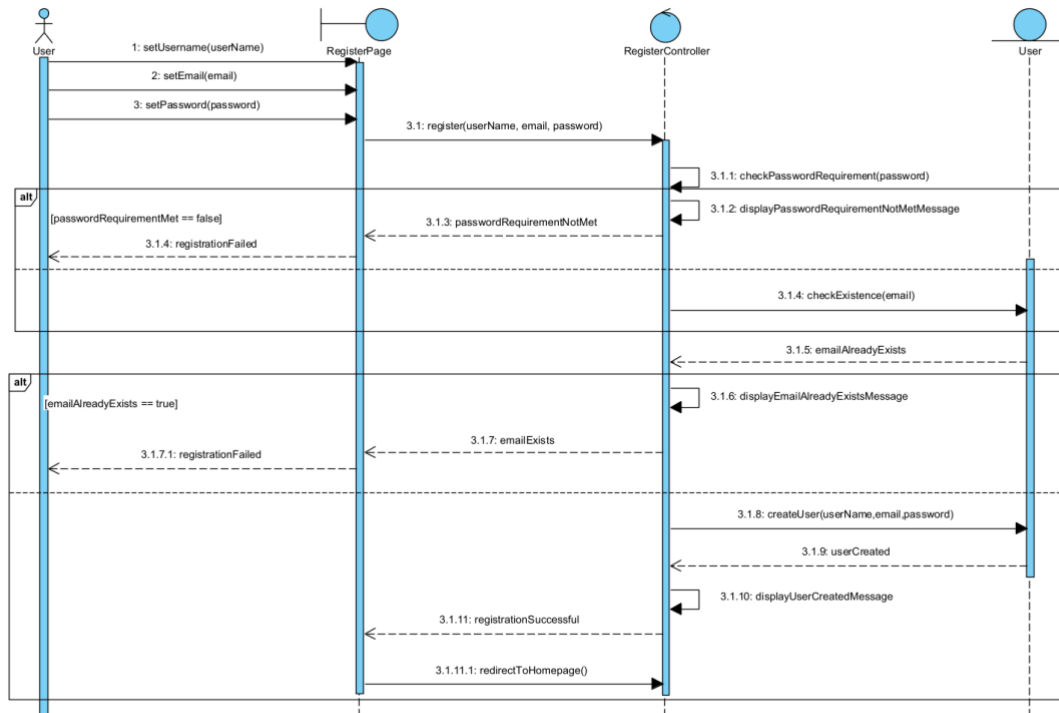
Login

sd [Login]

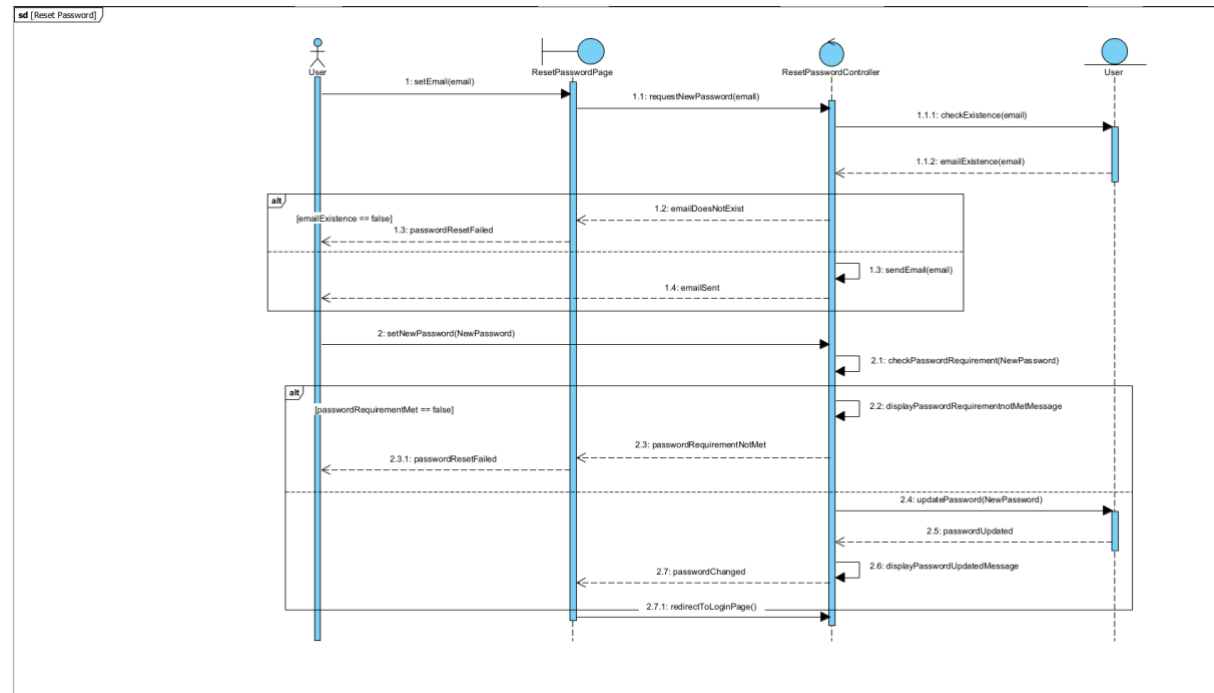


Register

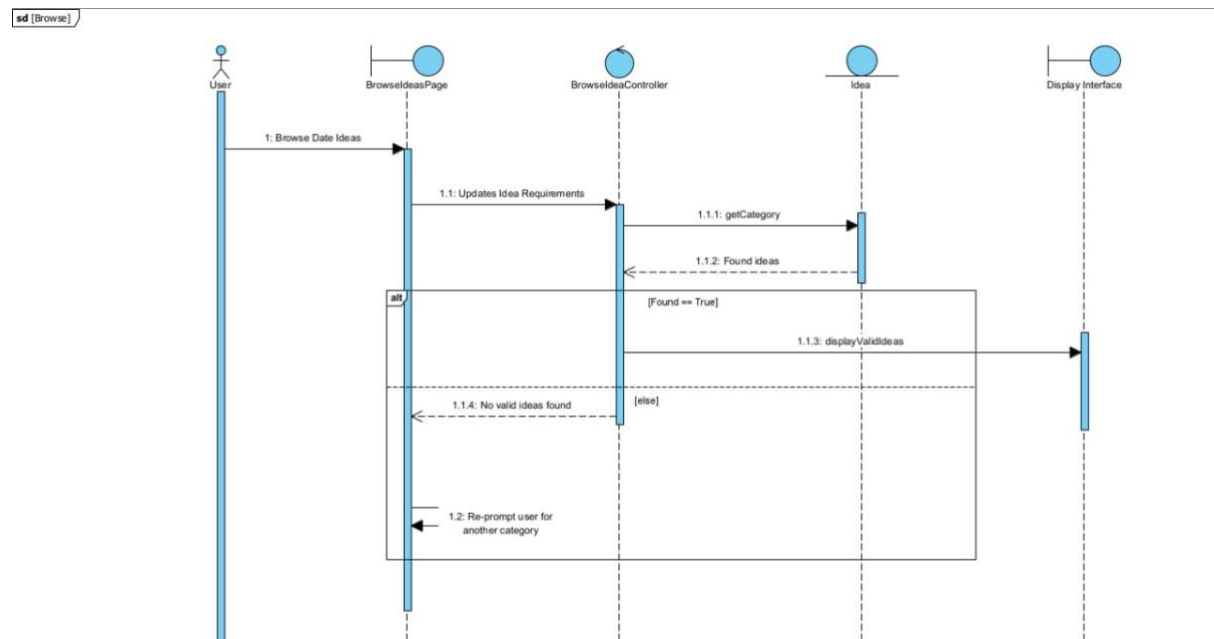
sd [Register]



Reset Password

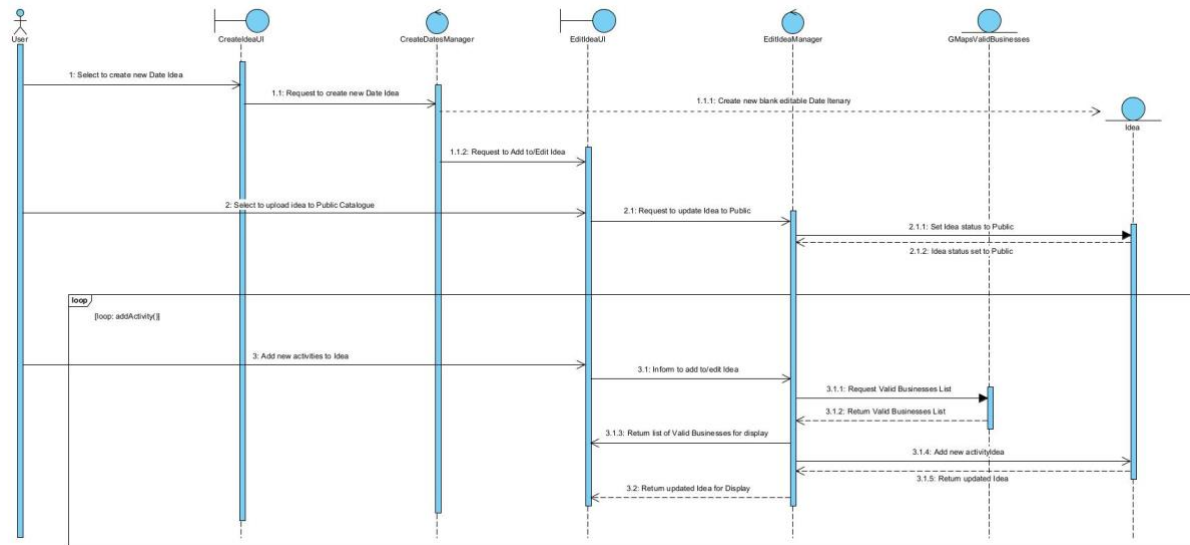


Browse Ideas



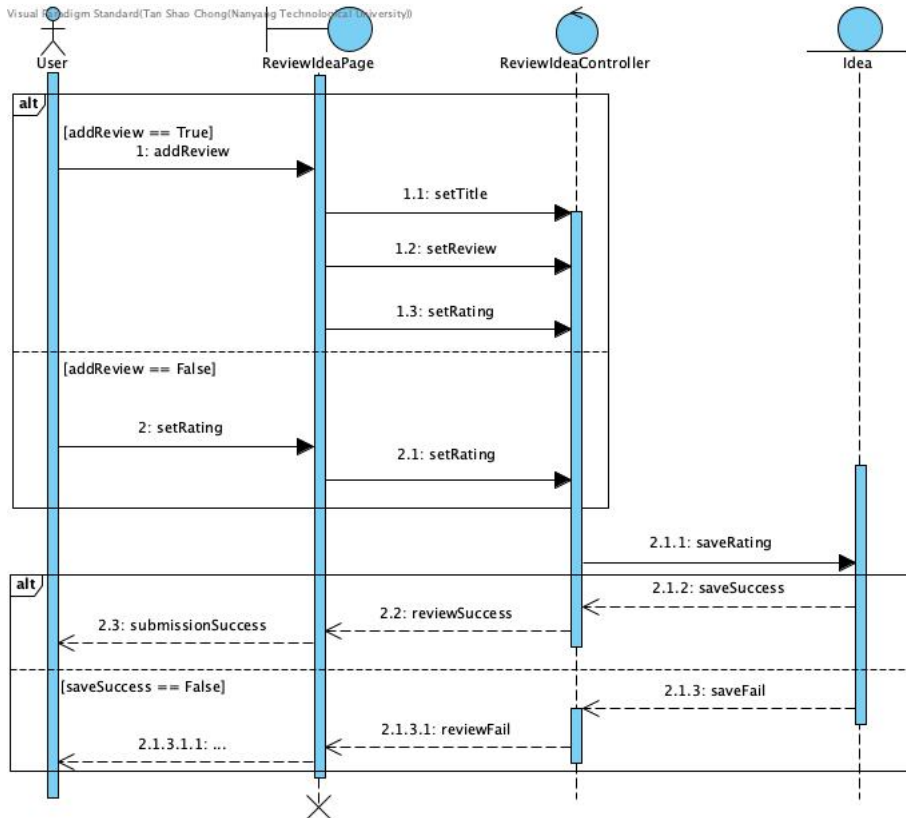
Create Idea

Visual Paradigm Standard (Tan Shao Chong (Nanyang Technological University))



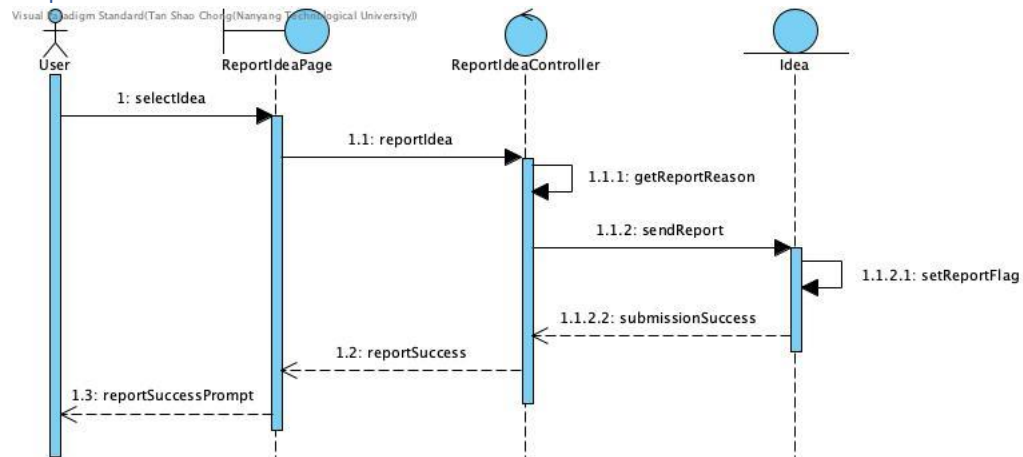
Review Idea

Visual Paradigm Standard (Tan Shao Chong (Nanyang Technological University))

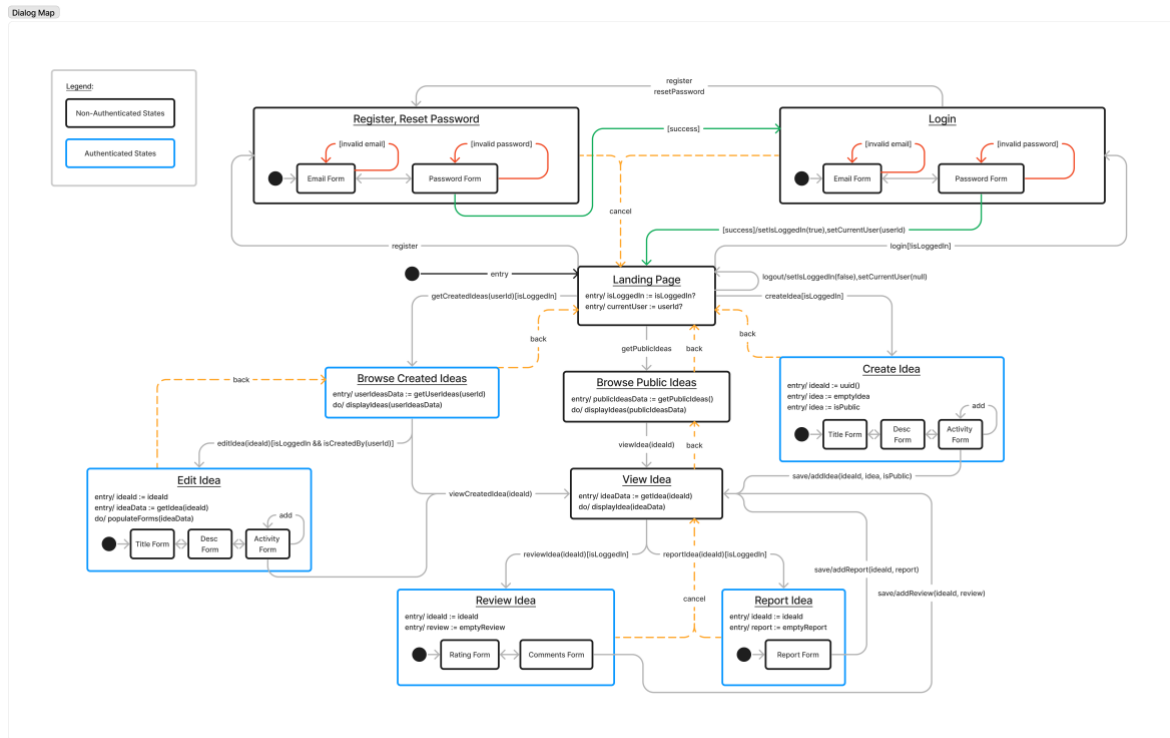


Report Idea

Visual Paradigm Standard (Tan Shao Chong (Nanyang Technological University))

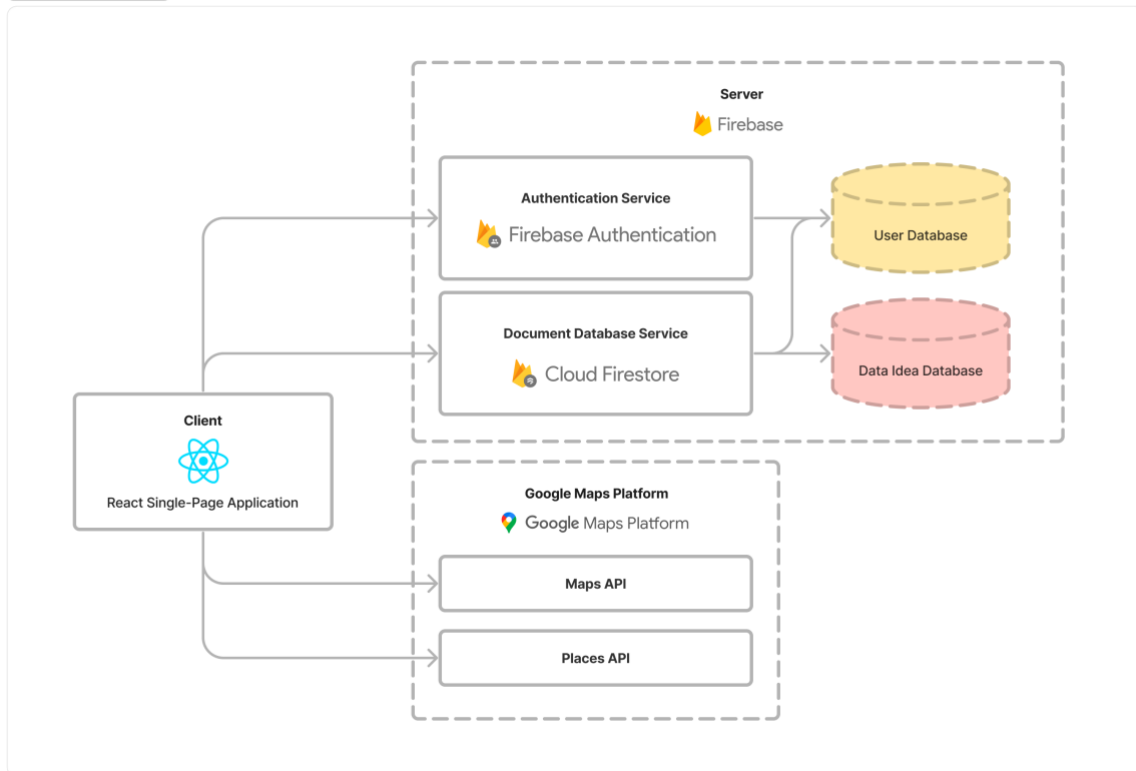


Dialog Map



System Architecture

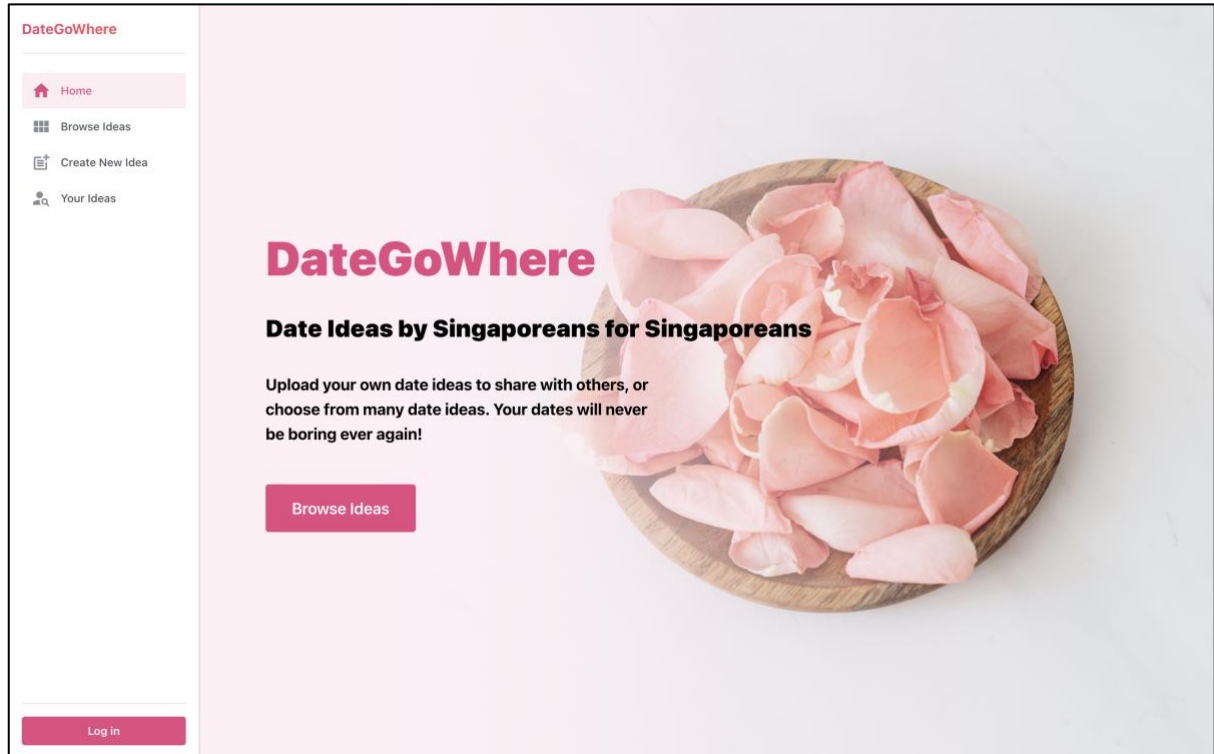
System Architecture



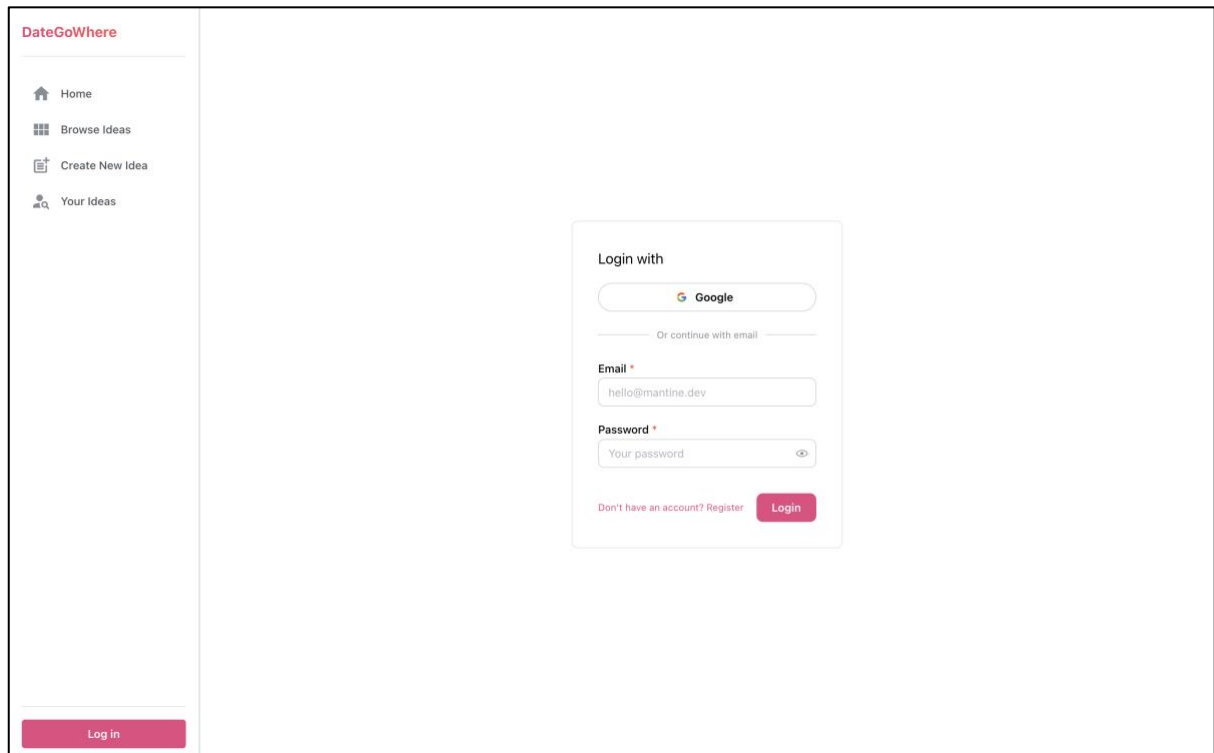
App Skeleton

(Project source code can be found in ~/app/client)

Landing



Authentication



Create Date Idea

DateGoWhere

Home

Browse Ideas

Create New Idea

Your Ideas

Tan Shao Chong

Manage account

Logout

Give your date a name! *

Romantic Night by The Bay

Start Time *

05:00 PM

End Time *

06:00 PM

Activity Name *

Picnic

Location *

Marina Gardens Drive, Marina Barrage, Singapore

Description *

Bring along your picnic kit!

Date Tags

Outdoor X Pick Category

Budget *

\$2

Discard

Save

+ Add activity

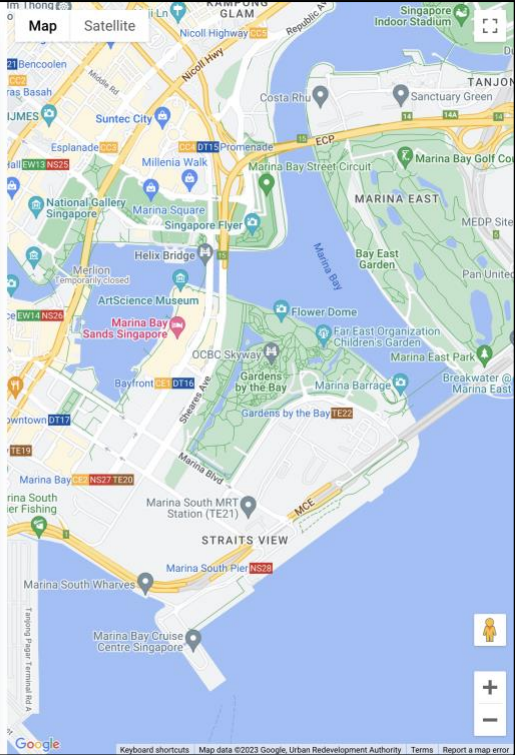
Save Draft

Get Route

Submit

Map

Satellite



Keyboard shortcuts Map data ©2023 Google Urban Redevelopment Authority Terms Report a map error

DateGoWhere

Home

Browse Ideas

Create New Idea

Your Ideas

Tan Shao Chong

Manage account

Logout

Give your date a name! *

Romantic Night by The Bay

17:00 - 19:00

Picnic @ Marina Barrage

18:00 - 19:00

Dinner @ Satay by the Bay

20:00 - 21:00

Sightseeing @ Gardens by the Bay

+ Add activity

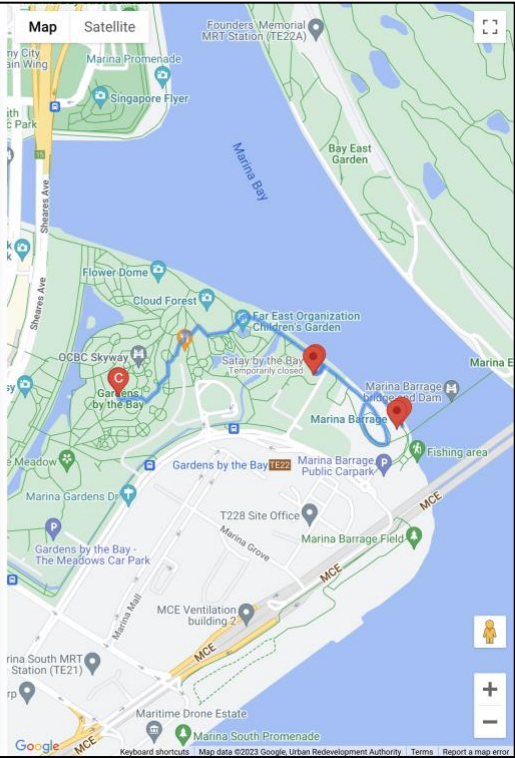
Save Draft

Get Route

Submit

Map

Satellite



Keyboard shortcuts Map data ©2023 Google Urban Redevelopment Authority Terms Report a map error

View Date Idea

DateGoWhere

Home

Browse Ideas

Create New Idea

Your Ideas

Tan Shao Chong

Manage account

Logout

Romantic Night by The Bay

BUDGET: \$45

DURATION: 4 HRS

5

17:00 - 18:00

Picnic @ Marina Barrage

18:00 - 19:00

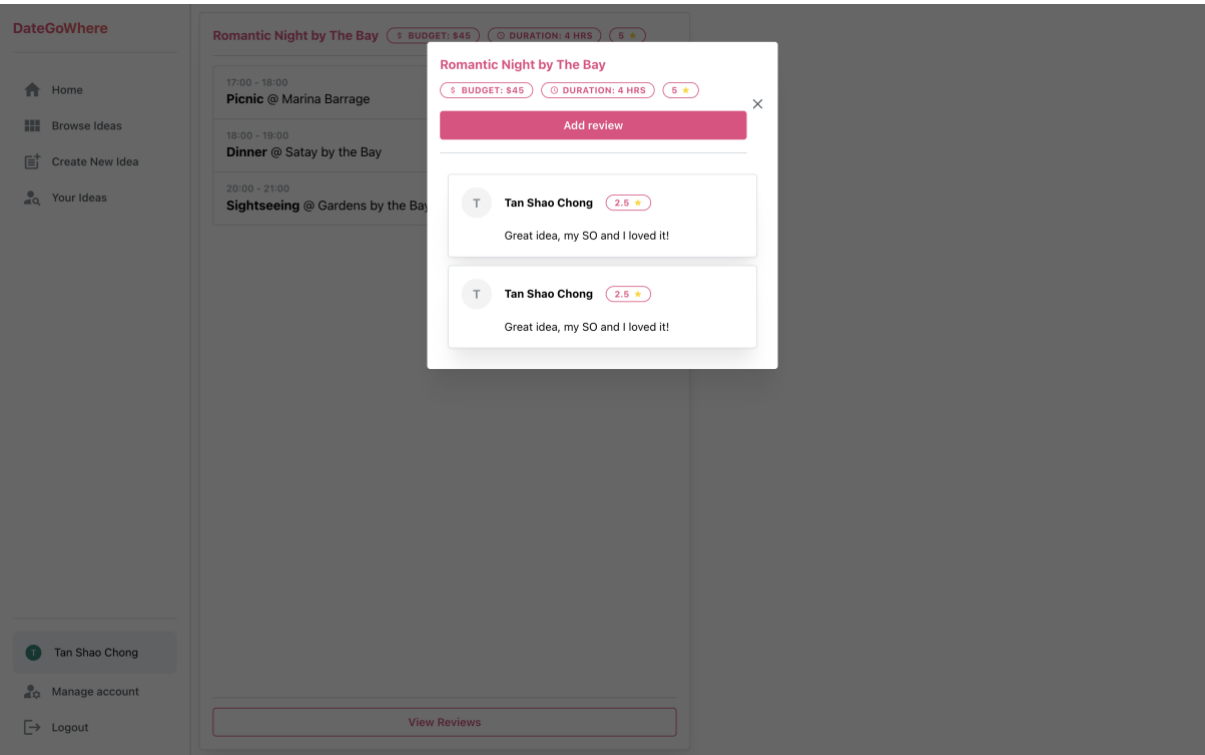
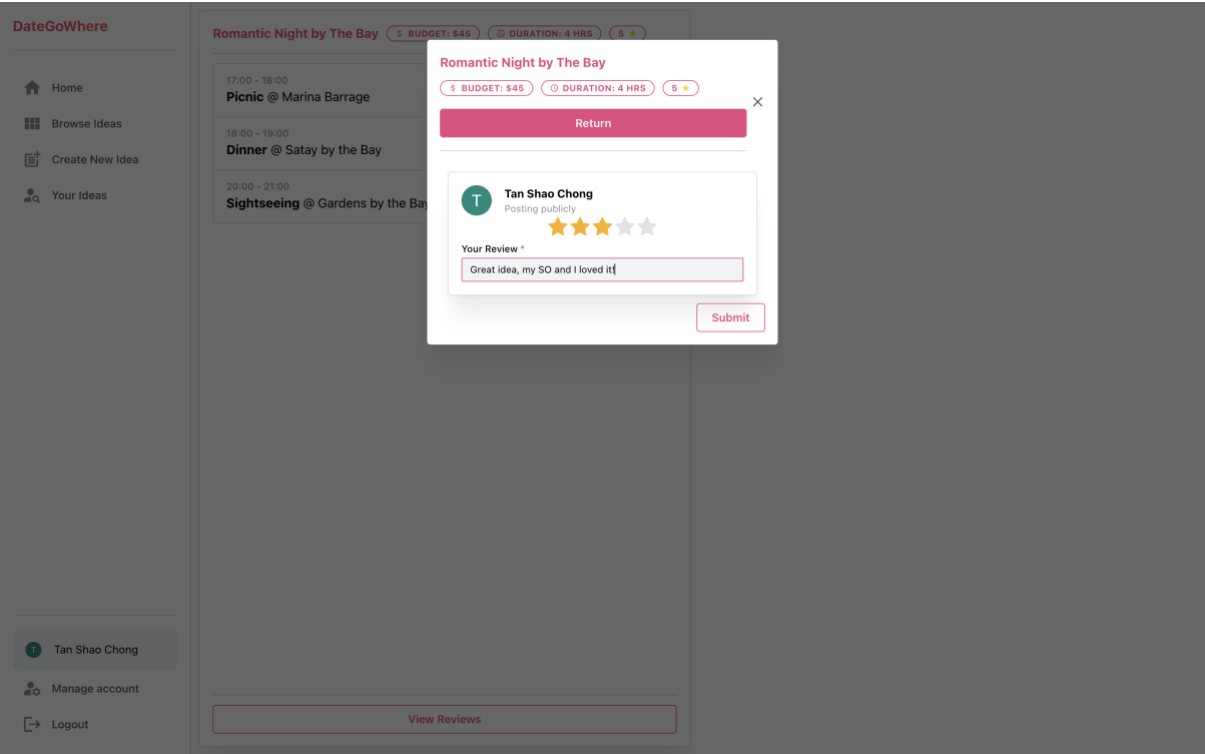
Dinner @ Satay by the Bay

20:00 - 21:00

Sightseeing @ Gardens by the Bay

View Reviews

Review Date Ideas



Browse Date Idea

DateGoWhere

Home

Browse Ideas

Create New Idea

Your Ideas

Tan Shao Chong

Manage account

Logout

Search by keywords

Q North, South, Park, etc.

Romantic Night by The Bay

Romantic Night by The Bay

Bring along your picnic kit!

PERFECT FOR YOU, IF YOU ENJOY

Show details